Polynomial Time Algorithms for Computing Approximate SVM Solutions with Guaranteed Accuracy

Los Alamos
National Laboratory

Don Hush, Clint Scovel, and Ingo Steinwart
Computer Research and Applications Group, CCS-3
Mail Stop B265
Los Alamos National Laboratory
Los Alamos, NM 87545

{dhush,jcs,ingo}@lanl.gov
505-665-2722

LANL Technical Report: LA–UR–05-7738

Report Date: September 16, 2005

Abstract

We describe a support vector machine (SVM) classifier design algorithm called L1–SVMD that produces approximate solutions with guaranteed accuracy and runs in polynomial time. This algorithm computes an approximate solution to the L1–SVM quadratic programming (QP) problem using a two stage approach where the first stage uses Simon's decomposition algorithm [8, 24, 16] to compute an approximate solution to a dual QP, and the second stage constructs an approximate primal solution from the approximate dual solution. For the second stage we establish a general method for constructing primal solutions with accuracy ϵ_p from dual solutions with accuracy ϵ_d by solving a set of *converse dual equations*, and then we design an efficient algorithm that approximately solves these equations to yield approximate primal solutions with accuracy $\epsilon_p \propto \sqrt{\epsilon_d}$. For the L1–SVMD algorithm we develop a run time bound that depends on the regularization parameter λ, the accuracy ϵ_p, the number of training samples n, and the kernel.

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W–7405–ENG–36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy.

1 Introduction

We begin with a formal definition of the classification problem. Let X be a pattern space and $Y := \{-1, 1\}$ be the label space. A classifier uses a decision function $h : X \to \mathbb{R}$ to assign the label sign $h(x)$ to every $x \in X$ (where sign $0 := 1$). Let P be an (unknown) probability measure on $X \times Y$. The performance of a decision function h is measured by its risk (average classification error)

$$\mathcal{R}_P(h) := P(\{(x, y) : \text{sign } h(x) \neq y\}).$$

The smallest achievable risk is the Bayes risk

$$\mathcal{R}_P^* := \inf\{\mathcal{R}_P(h) \mid h : X \to \mathbb{R} \text{ measurable}\}.$$

Let $T = ((x_1, y_1), (x_2, y_2), ..., (x_n, y_n)) \in (X \times Y)^n$ be a collection of n i.i.d. data samples drawn according to P. In the classification problem we seek a procedure that accepts T and produces a decision function h_T whose risk $\mathcal{R}_P(h_T)$ is close to the Bayes risk. More generally we seek a computationally efficient procedure that can accomplish this task for a large fraction of the training sets T drawn from distributions P that belong to a large class of distributions.

SVM classifiers map from the pattern space X to a Hilbert space H and implement a linear classifier in H. In particular, for a map $\Phi : X \to H$, SVM decision functions take the form

$$h_{\psi,b}(x) \ := \ \psi \cdot \Phi(x) + b$$

where $(\psi, b) \in H \times \mathbb{R}$ and $\cdot$ is the Hilbert space inner product. Since a large (possibly infinite) dimensional Hilbert space may be required for good performance a kernel function $k : X \times X \to \mathbb{R}$ is used to compute the Hilbert space inner product, i.e. $k(x_1, x_2) = \Phi(x_1) \cdot \Phi(x_2), \forall x_1, x_2 \in X$. In practice, rather than choose the map Φ, we often choose the kernel function k. Once a kernel has been chosen the classifier parameters (ψ_T, b_T) are determined by solving a convex quadratic programming (QP) problem parameterized by T. Several forms have been proposed for this QP problem, but few currently possess both favorable performance bounds (relative to the Bayes risk) and practical algorithms that are proven to run in polynomial–time. One that does is based on the QP problem

$$P_{SVM}: \qquad \begin{array}{rl} \min_{\psi,\xi,b} & \lambda\|\psi\|^2 + \sum_{i=1}^n u_i\xi_i \\ 1 - y_i(\psi \cdot \Phi(x_i) + b) - \xi_i \leq 0, & i = 1, ..., n \\ -\xi_i \leq 0, & i = 1, ..., n \end{array} \tag{1}$$

where $\lambda > 0$, $u_i > 0$ and $\sum_i u_i = 1$. In particular the so–called L1–SVM [26] sets $u_i = 1/n$, $i = 1, ..., n$, giving rise to the (P_{L1-SVM}) variant that is used to design parameters for the classification problem above. In contrast the DLD–SVM [25] sets

$$u_i = \begin{cases} \beta/n_1, & y_i = 1 \\ (1-\beta)/n_{-1}, & y_i = -1 \end{cases}$$

where $0 \leq \beta \leq 1$, giving rise to the $(P_{DLD-SVM})$ variant that is used to solve a density level detection problem for anomaly detection. Although our primary interest is in the (P_{L1-SVM}) variant, much of our intermediate analysis is carried out for the more general (P_{SVM}) problem.

Numerous algorithms have been proposed for (P_{SVM}), and while several are considered practical, few are known to possess polynomial run–time bounds. This problem presents some interesting computational challenges in that standard algorithms for convex QPs are not practical because

(P_{SVM}) may be very large (even infinite dimensional). For example the ellipsoid–based polynomial–time algorithm described by Bern and Eppstein is not practical because it requires an explicit representation of the data in H [3]. The so–called Wolfe Dual of (P_{SVM}) is given by

$$\acute{D}_{SVM}: \quad \begin{array}{r} \max_\alpha -\frac{1}{2}\langle Q\alpha,\alpha\rangle + \alpha\cdot 1 \\ \alpha\cdot y = 0 \\ 0\le\alpha\le u \end{array} \tag{2}$$

where

$$Q_{i,j} = y_i y_j \Phi(x_i)\cdot\Phi(x_j)/2\lambda, \quad 1\le i\le n, 1\le j\le n.$$

Since this problem is smaller (and always finite dimensional) it seems natural to consider an approach that first computes a solution α_T to $(\acute{D}_{SVM})$ and then uses α_T to compute a solution (ψ_T, b_T, ξ_T) to (P_{SVM}). In particular this might be accomplished by using α_T to form a Kuhn–Tucker vector for (P_{SVM}), establishing a set of optimality conditions expressed in terms of this Kuhn–Tucker vector using Theorem 28.1 in Rockafeller [22], and then formulating an optimization problem whose solutions satisfy these conditions. For example, given α_T, the optimization problem

$$\begin{array}{lcr} \psi_T & = & \frac{1}{2\lambda}\sum_{i=1}^n \alpha_{Ti} y_i \Phi(x_i) \\ (b_T,\xi_T) & \in & \arg\min_{b,\xi}\sum_{i=1}^n \xi_i \\ & & 1 - y_i(\psi_T\cdot\Phi(x_i)+b) - \xi_i \le 0, \quad i=1,...,n \\ & & -\xi_i \le 0, \quad i=1,...,n \end{array} \tag{3}$$

reduces the task of solving (P_{SVM}) to the task of solving an $(n+1)$–dimensional linear programming (LP) problem. However, $(\acute{D}_{SVM})$ can still be quite large. Indeed, simply specifying a problem instance requires order n^2 memory which, for medium to large values of n, is impractical for most modern computers. This issue has been addressed by employing algorithms that solve the Wolfe Dual QP problem by solving a sequence of smaller problems where each of the smaller QP problems is obtained by fixing a subset of the variables and optimizing with respect to the remaining variables. Algorithmic strategies that solve a QP problem in this way are called *decomposition* algorithms and a number have been developed for the Wolfe Dual QP problem [1, 4, 5, 7, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 24, 27], but few have been proven to run in polynomial time. Balcázar et al. present a randomized decomposition algorithm whose expected run time is $O\big((n+r(\mathtt{k}^2d^2))\,\mathtt{k}d\log n\big)$ where n is the number of samples, d is the dimension of the input space, $1\le\mathtt{k}\le n$ is a data dependent parameter and $r(\mathtt{k}^2d^2)$ is the run time required to exactly solve the Wolfe Dual QP over $\mathtt{k}^2d^2$ samples [1]. This algorithm is attractive when $\mathtt{k}^2d^2 \ll n$, but in practice the value of $\mathtt{k}$ is unknown and it may be large when the Bayes risk is not close to zero. Hush and Scovel [9] define a class of *rate certifying algorithms* that are guaranteed to produce an approximate solution to the Wolfe Dual QP in polynomial time. The key to developing a successful rate certifying algorithm is in the method used to determine the *working sets*, which are the subsets of variables to be optimized at each iteration. Currently the fastest guaranteed rates are obtained with Simon's working set selection algorithm [24] which yields a rate certifying decomposition algorithm that produces an ϵ_d–optimal solution to the Wolfe dual in $O\left(\frac{nK}{\lambda\epsilon_d} + n^2\ln\frac{\lambda n}{K}\right)$ time where $K = \max_i k(x_i,x_i)$ [8, 16].

Existing computational guarantees hold only for *approximate* solutions to a *dual QP problem* [8, 16]. We remedy this situation by establishing a framework for the accurate construction of approximate primal solutions from approximate dual solutions, designing and analyzing a specific construction method, ans designing and analyzing a computationally efficient algorithm for this method. In particular Section 2 establishes a framework for constructing approximate solutions to (P_{SVM}) with accuracy ϵ_p from approximate solutions to $(\acute{D}_{SVM})$ with accuracy ϵ_d by solving a set of *converse dual equations*. We show that exact solutions to these equations yield approximate primal solutions with $\epsilon_p = 4\epsilon_d$, but we are unable to produce a computationally efficient algorithm to exactly solve these equations. On the other hand we determine a computationally efficient method for approximately solving these equations that yields approximate primal solutions with $\epsilon_p \propto \sqrt{\epsilon_d}$. In Section 3 we combine this method with Simon's rate certifying algorithm for the dual [8, 24, 16] to form a learning algorithm for the L1–SVM variant of (P_{SVM}) that is polynomial in n, $1/\lambda$, and $1/\epsilon_p$.

Remark 1 *Before we proceed let us clarify the scope of our run time analysis. The types of computations required to design and implement SVMs include* `constant time operations` *(such as addition, multiplication and comparison) and* `kernel evaluations` *whose computational requirements are almost never constant time and vary significantly from one kernel to the next. In this paper we describe an algorithm that requires a polynomial number of constant time operations and a polynomial number of kernel evaluations.*

2 Constructing an Approximate Solution to the Primal

An efficient procedure exists for determining an *exact* primal solution from an *exact* dual solution by solving the LP problem in (3) (e.g. such a procedure is described in the last part of the proof of Theorem 3 in Section 3), but a suitable method for determining an *approximate* primal solution from an *approximate* dual solution is currently unknown. The analysis in this section suggests that the latter involves a serious trade–off between computation and accuracy. We briefly investigate this issue and then analyze an efficient procedure that closely resembles current practice.

We start by defining approximate solution sets for maximization problems. The definition for minimization problems is analogous.

Definition 1 *Let (P) be a maximization problem with domain Θ and criterion function $G : \Theta \to \bar{\mathbb{R}}$. Let $G^* := \sup_{\theta\in\Theta} G(\theta)$ be its optimal value. Then for any $0 \le \epsilon < \infty$ we define*

$$\mathcal{O}_\epsilon(P) \ := \ \{\theta \in \Theta : G(\theta) \ge G^* - \epsilon\}$$

to be the set of ϵ–maximizers of (P).

To simplify notation we use the variable z to represent points mapped from X, i.e. $z := \Phi(x)$. The criterion function for (P_{SVM}) is

$$G_P(\psi, b, \xi) = \lambda\|\psi\|^2 + \sum_{i=1}^n u_i\xi_i + \delta_{S_1}(\psi, b, \xi)$$

where δ_S is the indicator function of the set S,

$$\delta_S(\theta) \ := \ \begin{cases} 0, & \theta \in S \\ \infty, & \theta \notin S \end{cases}$$

and $S_1 = \{(\psi, b, \xi) : 1 - y_i(\psi \cdot z_i + b) - \xi_i \le 0 \text{ and } \xi_i \ge 0, \ i = 1, ..., n\}$. The Lagrangian [22] for (P_{SVM}) is

$$L(\psi, \xi, b; \alpha, \mu) = \\ \begin{cases} \lambda\|\psi\|^2 + \sum_{i=1}^n u_i\xi_i - \sum_{i=1}^n \alpha_i\big(y_i(\psi \cdot z_i + b) - 1 + \xi_i\big) - \sum_{i=1}^n \mu_i\xi_i & (\alpha, \mu) \in E \\ -\infty & (\alpha, \mu) \notin E \end{cases}$$

where $E = \{(\alpha,\mu) : \alpha_i \geq 0, \mu_i \geq 0, i = 1,...,n\}$. The Lagrange dual criterion function is

$$G_D(\alpha,\mu) \;:=\; \inf_{\psi,\xi,b} L(\psi,\xi,b;\alpha,\mu) =$$

$$\begin{cases} -\infty & \alpha_i + \mu_i \neq u_i \text{ for any } i \text{ or } (\alpha,\mu) \notin E \\ \inf_{\psi,b} \lambda\|\psi\|^2 - \sum_{i=1}^{n} \alpha_i\big(y_i(\psi\cdot z_i + b) - 1\big), & \alpha_i + \mu_i = u_i \ \forall i \text{ and } (\alpha,\mu) \in E. \end{cases}$$

Define the matrix Q

$$Q_{i,j} = y_i y_j z_i \cdot z_j / 2\lambda, \quad 1 \leq i \leq n, 1 \leq j \leq n.$$

Since

$$\inf_{\psi,b} \lambda\|\psi\|^2 - \sum_{i=1}^{n} \alpha_i\big(y_i(\psi\cdot z_i + b) - 1\big) = \begin{cases} -\infty, & \alpha\cdot y \neq 0 \\ \inf_{\psi} \lambda\|\psi\|^2 - \sum_{i=1}^{n} \alpha_i\big(y_i\psi\cdot z_i - 1\big), & \alpha \cdot y = 0 \end{cases}$$

and

$$\inf_{\psi} \lambda\|\psi\|^2 - \sum_{i=1}^{n} \alpha_i\big(y_i\psi\cdot z_i - 1\big) = -\frac{1}{2}\langle Q\alpha,\alpha\rangle + \alpha\cdot 1$$

we obtain that the Lagrangian dual criterion is

$$G_D(\alpha,\mu) = -\frac{1}{2}\langle Q\alpha,\alpha\rangle + \alpha\cdot 1 - \delta_{S_2}(\alpha) - \delta_{S_3}(\alpha,\mu) \tag{4}$$

where $S_2 = \{\acute{\alpha} : 0 \leq \acute{\alpha} \leq u \text{ and } \acute{\alpha}\cdot y = 0\}$ and $S_3 = \{(\acute{\alpha},\acute{\mu}) : \acute{\alpha} + \acute{\mu} = u\}$. Therefore we can now define the corresponding SVM dual optimization problem

$$D_{SVM}: \quad \begin{array}{r} \max_{\alpha,\mu} -\frac{1}{2}\langle Q\alpha,\alpha\rangle + \alpha\cdot 1 \\ 0 \leq \alpha \leq u \\ \alpha\cdot y = 0 \\ \alpha + \mu = u \end{array} \tag{5}$$

Let

$$\acute{G}_D(\alpha) = -\frac{1}{2}\langle Q\alpha,\alpha\rangle + \alpha\cdot 1 - \delta_{S_2}(\alpha)$$

denote the so-called Wolfe dual criterion function. We note that this function is not the true Wolfe dual, which is defined on $(\psi, b, \xi : \alpha, \mu)$ space [17], but it is equivalent to it. We conclude that

$$G_D(\alpha,\mu) = \acute{G}_D(\alpha) - \delta_{S_3}(\alpha,\mu)$$

and solving for $(\alpha^*,\mu^*) \in \arg\max G_D$ is equivalent to solving for $\alpha^* \in \arg\max \acute{G}_D$ and then setting $\mu^* = u - \alpha^*$.

The following theorem shows that approximate solutions of (P_{SVM}) can be constructed from approximate solutions to its dual (D_{SVM}) by solving a set of converse dual equations.

Theorem 1 *Let $(\alpha^*,\mu^*) \in \mathcal{O}_\epsilon(D_{SVM})$ and define the set of points*

$$CD_\sigma \;:=\; \left\{ (\psi,b,\xi) \in H \times \mathbb{R}^{1+n} : \begin{array}{cc} (a) & \xi_i \geq 0, \quad i = 1,...,n, \\ (b) & \xi_i \geq 1 - y_i(\psi\cdot z_i + b), \quad i = 1,...,n, \\ (c) & \sum_{i=1}^{n} u_i\xi_i + 2\lambda\psi\cdot\bar{\psi} - \alpha^*\cdot 1 \leq 2\sigma \\ (d) & \psi = \bar{\psi} + \acute{\psi} \\ (e) & \|\acute{\psi}\| \leq \sqrt{2\sigma/\lambda} \\ (f) & \bar{\psi} = \frac{1}{2\lambda}\sum_{i=1}^{n} \alpha_i^* y_i z_i \end{array} \right\} \tag{6}$$

that satisfy the σ–converse dual equations (a)–(f). Then CD_σ is nonempty for all $\sigma \geq \epsilon$ and for $\sigma \geq 0$ it satisfies $CD_\sigma \subseteq \mathcal{O}_{4\sigma}(P_{SVM})$.

Proof: Our proof is an application of [23, Corollary 2.7] as described in [23, Remark 2.8]. First we note that the assumptions in [23, Corollary 2.7] are met with ϵ_1 from that corollary equal to zero since an exact solution to (P_{SVM}) with nonnegative criterion value always exists and the duality gap is zero by Theorem 4 in the appendix. Referring to the notation in that paper, we identify the convex parameter space $C = H \times \mathbb{R}^{1+n}$ and

$$f_0(\psi, b, \xi) = \lambda\|\psi\|^2 + \sum_{i=1}^{n} u_i\xi_i$$

$$f_i(\psi, b, \xi) = 1 - y_i(\psi \cdot z_i + b) - \xi_i,\ i = 1, ..., n$$

$$f_{i+n}(\psi, b, \xi) = -\xi_i,\ i = 1, ..., n$$

where $m = r = 2n$. Then the set of points that satisfy the τ–Kuhn–Tucker conditions of Scovel et al. [23] become

$$KTC_\tau =$$

$$\left\{ (\psi, b, \xi, \alpha, \mu) \in C \times \mathbb{R}^{2n} : \begin{array}{ll} (a) & f_i(\psi, b, \xi) \leq 0,\ i = 1, ..., 2n \text{ and } \alpha \geq 0,\ \mu \geq 0 \\ (b) & -\tau \leq \sum_{i=1}^n \alpha_i f_i(\psi, b, \xi) + \sum_{i=1}^n \mu_i f_{n+i}(\psi, b, \xi) \\ (c) & \exists\ 0 \leq \sigma_0 \leq \tau \text{ such that} \\ & 0 \in \partial_{\sigma_0} f_0(\psi, b, \xi) + \sum_{i=1}^n \alpha_i df_i(\psi, b, \xi) + \sum_{i=1}^n \mu_i df_{n+i}(\psi, b, \xi) \end{array} \right\} \tag{7}$$

where df denotes the differential of a function f and ∂_τ denotes the τ–subdifferential operator [6]. Let

$$KTC_{\tau|\epsilon} := \{(\psi, b, \xi) \in C : \exists(\alpha, \mu) \in \mathcal{O}_\epsilon(D_{SVM}) \text{ such that } (\psi, b, \xi, \alpha, \mu) \in KTC_\tau\}.$$

According to [23, Corollary 2.7] the set $KTC_{\tau|\epsilon}$ is nonempty for all $\tau \geq 2\epsilon$ and for $\tau \geq 0$ it satisfies $KTC_{\tau|\epsilon} \subseteq \mathcal{O}_{2\tau}(P_{SVM})$. Thus to complete the proof it is sufficient to show that $CD_\sigma \subseteq KTC_{2\sigma|\epsilon}$, $\sigma \geq 0$.

Let us start by representing (7) at an approximate maximizer $(\alpha, \mu) = (\alpha^*, \mu^*) \in \mathcal{O}_\sigma(D_{SVM})$ and $\tau = 2\sigma$. Since $(\alpha^*, \mu^*) \in \mathcal{O}_\sigma(D_{SVM})$ it follows that (α^*, μ^*) must be feasible for (D_{SVM}), that is

$$\begin{array}{r} 0 \leq \mu^* \leq u \\ 0 \leq \alpha^* \leq u \\ \alpha^* \cdot y = 0 \\ \alpha^* + \mu^* = u \end{array} . \tag{8}$$

Therefore

$$\sum_{i=1}^{n} \alpha_i^* f_i(\psi, b, \xi) + \sum_{i=1}^{n} \mu_i^* f_{n+i}(\psi, b, \xi) = \alpha^* \cdot 1 - \sum_{i=1}^{n} u_i\xi_i - \psi \cdot \sum_{i=1}^{n} \alpha_i^* y_i z_i$$

and so with $\tau = 2\sigma$ and $\bar{\psi} = \frac{1}{2\lambda}\sum_{i=1}^n \alpha_i^* y_i z_i$ an equivalent set of equations to (7a)-(7c) is

$$\begin{array}{lc} (a) & \xi_i \geq 0, \quad i = 1, ..., n, \\ (b) & \xi_i \geq 1 - y_i(\psi \cdot z_i + b), \quad i = 1, ..., n, \\ (c) & \sum_{i=1}^n u_i\xi_i + 2\lambda\psi \cdot \bar{\psi} - \alpha^* \cdot 1 \leq 2\sigma \\ (d) & 0 \in \partial_{2\sigma}(\lambda\|\psi\|^2)(\psi, b, \xi) + d(\sum_{i=1}^n u_i\xi_i)(\psi, b, \xi) + \sum_{i=1}^n \alpha_i^* df_i(\psi, b, \xi) \\ & + \sum_{i=1}^n \mu_i^* df_{n+i}(\psi, b, \xi) \end{array} . \tag{9}$$

Thus it remains to show that (9a)-(9d) imply (6a)-(6f). To compute the 2σ subdifferential of the quadratic function we use Example XI.1.2.2 (p. 95) in Hiriart-Urruty and Lemaréchal [6] to obtain

$$\partial_{2\sigma}\left(\lambda\|\psi\|^2\right)(\psi,b,\xi)=\left\{2\lambda\begin{bmatrix}\psi+\acute{\psi}\\0\\0\end{bmatrix}:\|\acute{\psi}\|^2\leq 2\sigma/\lambda\right\}.$$

We also have

$$d\left(\sum_{i=1}^{n}u_i\xi_i\right)(\psi,b,\xi)=\begin{bmatrix}0\\0\\u\end{bmatrix},$$

$$df_i(\psi,b,\xi)=-\begin{bmatrix}y_iz_i\\y_i\\e_i\end{bmatrix},i=1,..,n$$

and

$$df_i(\psi,b,\xi)=-\begin{bmatrix}0\\0\\e_i\end{bmatrix},i=n+1,..,2n$$

where $e_i=(0,...,1,...,0)$ is an n–tuple with a 1 in position i and 0 in all other positions. Thus we conclude that the equations (9a)-(9d) imply (6a)-(6f) so the proof is finished. ∎

Theorem 1 says that we can construct a 4ϵ–minimizer of (P_{SVM}) from an ϵ–maximizer of (D_{SVM}) by employing an algorithm that solves the ϵ–converse dual equations. We now seek such an algorithm. One approach is to formulate an efficiently solvable programming problem whose solutions satisfy the ϵ–converse dual equations. This is relatively straightforward when an *exact* dual solution (α_T,μ_T) is available. For example it is easy to verify that solutions to (3) satisfy the ϵ–converse dual equations with $\epsilon=0$ and so any algorithm that solves the LP problem in (3) will give an exact solution to the primal. Furthermore any algorithm that substitutes an exact solution α_T into (3) and then computes an ϵ–minimizer (b^*,ξ^*) for the LP problem gives an ϵ–minimizer (ψ_T,b^*,ξ^*) to the primal. However when only an *approximate* dual solution is available it appears to be more difficult to determine an efficiently solvable programming problem whose solutions satisfy the ϵ–converse dual equations. On one hand we can formulate many programming problems whose (exact) solutions would yield solutions to the ϵ–converse dual equations. For example any solution of the QP problem

$$\begin{aligned}\min\nolimits_{\psi,b,\xi}\ &\|\psi-\bar{\psi}\|^2\\ \xi_i\geq 0,\quad &i=1,...,n,\\ \xi_i\geq 1-y_i(\psi\cdot z_i+b),\quad &i=1,...,n,\\ \textstyle\sum_{i=1}^{n}u_i\xi_i+2\lambda\psi\cdot\bar{\psi}-\alpha^*\cdot 1\leq 2\sigma&\\ \bar{\psi}=\tfrac{1}{2\lambda}\textstyle\sum_{i=1}^{n}\alpha_i^*y_iz_i&\end{aligned}\tag{10}$$

satisfies (6). Furthermore with $\sigma=\epsilon$ a solution is guaranteed. Another QP problem with the same property is

$$\begin{aligned}\min\nolimits_{\psi,b,\xi}\textstyle\sum_{i=1}^{n}u_i\xi_i+2\lambda\psi\cdot\bar{\psi}&\\ \xi_i\geq 0,\quad &i=1,...,n,\\ \xi_i\geq 1-y_i(\psi\cdot z_i+b),\quad &i=1,...,n,\ .\\ \|\psi-\bar{\psi}\|^2\leq 2\sigma/\lambda&\\ \bar{\psi}=\tfrac{1}{2\lambda}\textstyle\sum_{i=1}^{n}\alpha_i^*y_iz_i&\end{aligned}\tag{11}$$

On the other hand both of these, like the primal problem, may be very large and therefore cannot be solved directly. This will be true for any programming problem that optimizes over the RKHS, but it may be difficult to avoid optimizing over this space since the converse dual equations include a variable that lives in this space. One possibility is to fix this variable, e.g. $\acute{\psi} = 0$, and then formulate a programming problem over the remaining variables. For example we might consider the simpler optimization problem

$$\begin{array}{r} \min_{\psi=\bar{\psi},b,\xi} \sum_{i=1}^{n} u_i\xi_i \\ \xi_i \geq 0, \quad i = 1, ..., n, \\ \xi_i \geq 1 - y_i(\bar{\psi} \cdot z_i + b), \quad i = 1, ..., n, \\ \bar{\psi} = \frac{1}{2\lambda}\sum_{i=1}^{n} \alpha_i^* y_i z_i \end{array} . \tag{12}$$

This is a realization of (3) for approximate dual solutions and is therefore a natural problem to consider. It is an open question whether solutions to this problem satisfy (6), but we suspect not. However the following theorem shows that this last problem can be simplified to a one dimensional convex optimization problem whose approximate solutions are approximately optimal for the primal but with a larger degree of suboptimality.

Theorem 2 *Consider the SVM programming problem (P_{SVM}) with $|z_i|^2 \leq K, i = 1, .., n$. Suppose that $(\alpha^*, \mu^*) \in \mathcal{O}_\epsilon(D_{SVM})$ and $G_D(\alpha^*, \mu^*) \geq 0$. Let*

$$\bar{\psi} = \frac{1}{2\lambda}\sum_{i=1}^{n} \alpha_i^* y_i z_i \tag{13}$$

$$\bar{\xi}_i(b) = \max\left(0, 1 - y_i(\bar{\psi} \cdot z_i + b)\right),\ i = 1, .., n \tag{14}$$

and let (P_{offset}) be the one dimensional optimization problem

$$\min_b \sum_{i=1}^{n} u_i \bar{\xi}_i(b).$$

If $\bar{b} \in \mathcal{O}_{\epsilon_o}(P_{offset})$ then $(\bar{\psi}, \bar{b}, \bar{\xi}(\bar{b})) \in \mathcal{O}_{\epsilon_p}(P_{SVM})$ with $\epsilon_p = 2\epsilon_o + 4\epsilon + 4\sqrt{\epsilon}\big(\sqrt{2} + \sqrt{K/2\lambda}\big)$.

Proof: We start by replacing the variables ξ_i in the converse dual equations with

$$\xi_i(\psi, b) \ := \ \max\left(0, 1 - y_i(\psi \cdot z_i + b)\right), \quad i = 1, ..., n,$$

and then simplifying to obtain a new set of equations described by the following lemma.

Lemma 1 *Let $(\alpha^*, \mu^*) \in \mathcal{O}_\epsilon(D_{SVM})$ and define the set*

$$MDC_\sigma \ := \ \left\{ (\psi, b, \xi) \in H \times \mathbb{R}^{1+n} : \begin{array}{cc} (\acute{a}) & \sum_{i=1}^{n} u_i\xi_i(\psi, b) + 2\lambda\psi \cdot \bar{\psi} - \alpha^* \cdot 1 \leq 2\sigma \\ (\acute{b}) & \psi = \bar{\psi} + \acute{\psi} \\ (\acute{c}) & \|\acute{\psi}\| \leq \sqrt{2\sigma/\lambda} \\ (\acute{d}) & \bar{\psi} = \frac{1}{2\lambda}\sum_{i=1}^{n} \alpha_i^* y_i z_i \end{array} \right\} \tag{15}$$

of points that satisfy the σ–modified converse dual equations $(\acute{a})$–$(\acute{d})$. If $(\psi, b, \xi) \in CD_\sigma$ then $(\psi, b, \xi(\psi, b)) \in CD_\sigma$ and $(\psi, b) \in MDC_\sigma$. Conversely if $(\psi, b) \in MDC_\sigma$ then $(\psi, b, \xi(\psi, b)) \in CD_\sigma$.

Proof: The proof follows trivially from the monotonicity in ξ of the converse dual equations (6). ■

Now we choose $\acute{\psi} = 0$ so that equation ($\acute{c}$) is satisfied for any $\sigma \geq 0$ and we use the fact that α^* is an ϵ–maximizer of the dual to determine a value of σ for which equation ($\acute{a}$) is satisfied. Since (α^*, μ^*) is an ϵ–maximizer it must be feasible. The assumptions give

$$G(\alpha^*, \mu^*) = -\frac{1}{2}\langle Q\alpha^*, \alpha^*\rangle + \alpha^* \cdot 1 \geq 0.$$

However since $\frac{1}{2}\langle Q\alpha^*, \alpha^*\rangle = \lambda|\bar{\psi}|^2$ we obtain

$$\lambda|\bar{\psi}|^2 - \alpha^* \cdot 1 \leq 0$$

and since $\alpha^* \cdot 1 \leq 1$ (because $0 \leq \alpha_i \leq u_i$ and $u \cdot 1 = 1$) we obtain

$$|\bar{\psi}| \leq \sqrt{1/\lambda}. \tag{16}$$

We know from Theorem 1 that there exists a $(\psi^*, \xi^*, b^*) \in CD_\epsilon$ and by Lemma 1 $(\psi^*, b^*) \in MDC_\epsilon$. If we denote

$$\xi_i^*(b) = \max(0, 1 - y_i(\psi^* \cdot z_i + b)), \quad i = 1, ..., n,$$

then it follows from the identity

$$|\max(0, s) - \max(0, t)| \leq |s - t|,$$

and the inequality $\|\psi^* - \bar{\psi}\| \leq \sqrt{2\epsilon/\lambda}$ that

$$|\bar{\xi}_i(b^*) - \xi_i^*(b^*)| \leq |(\psi^* - \bar{\psi}) \cdot z_i| \leq \sqrt{2\epsilon K/\lambda}, \quad i = 1, ..., n.$$

In addition it follows that

$$\begin{aligned}\sum_{i=1}^n u_i\bar{\xi}_i(b^*) + 2\lambda\bar{\psi}\cdot\bar{\psi} - \alpha^*\cdot 1 &= \sum_{i=1}^n u_i\xi_i^*(b^*) + 2\lambda\psi^*\cdot\bar{\psi} - \alpha^*\cdot 1 + \sum_{i=1}^n u_i(\bar{\xi}_i(b^*) - \xi_i^*(b^*)) + 2\lambda(\bar{\psi} - \psi^*)\cdot\bar{\psi} \\ &\leq 2\epsilon + \sqrt{2\epsilon K/\lambda} + 2\lambda\|\psi^* - \bar{\psi}\|\|\bar{\psi}\| \\ &\leq 2\epsilon + 2\sqrt{\epsilon}\big(\sqrt{2} + \sqrt{K/2\lambda}\big).\end{aligned}$$

Therefore with $\bar{b} \in \mathcal{O}_{\epsilon_o}(P_{offset})$ and $\bar{b}_0 \in \mathcal{O}_0(P_{offset})$ we have

$$\begin{aligned}\sum_{i=1}^n u_i\bar{\xi}_i(\bar{b}) + 2\lambda\bar{\psi}\cdot\bar{\psi} - \alpha^*\cdot 1 &\leq \sum_{i=1}^n u_i\bar{\xi}_i(\bar{b}_0) + \epsilon_o + 2\lambda\bar{\psi}\cdot\bar{\psi} - \alpha^*\cdot 1 \\ &\leq \sum_{i=1}^n u_i\bar{\xi}_i(b^*) + \epsilon_o + 2\lambda\bar{\psi}\cdot\bar{\psi} - \alpha^*\cdot 1 \\ &\leq \epsilon_o + 2\epsilon + 2\sqrt{\epsilon}\big(\sqrt{2} + \sqrt{K/2\lambda}\big).\end{aligned}$$

Since $\bar{\xi}(\bar{b}) = \xi(\bar{\psi}, \bar{b})$ we conclude that

$$\sum_{i=1}^n u_i\xi_i(\bar{\psi}, \bar{b}) + 2\lambda\bar{\psi}\cdot\bar{\psi} - \alpha^*\cdot 1 \leq \epsilon_o + 2\epsilon + 2\sqrt{\epsilon}\big(\sqrt{2} + \sqrt{K/2\lambda}\big).$$

Consequently, $(\bar{\psi}, \bar{b})$ satisfies the modified converse dual equations with $\sigma = \frac{\epsilon_o}{2} + \epsilon + \sqrt{\epsilon}\big(\sqrt{2} + \sqrt{K/2\lambda}\big)$ and by Lemma 1 we know that $(\bar{\psi}, \bar{b}, \xi(\bar{\psi}, \bar{b}))$ satisfies the converse dual equations. Theorem 1 now implies the result. ■

In the next section we describe a polynomial time algorithm that uses Theorem 2 to compute an approximate solution to (P_{SVM}). Since this algorithm computes an exact solution to (P_{offset}) the accuracy requirement for the dual solution is given by the following corollary.

Corollary 1 *Consider the SVM programming problem (P_{SVM}) with $\lambda > 0$ and $|z_i|^2 \leq K, i = 1,..,n$. To produce an ϵ_p–minimizer of (P_{SVM}) from an ϵ_d–maximizer of (D_{SVM}) using (13)-(14) and $\bar{b} \in \mathcal{O}_0(P_{offset})$ it is sufficient that*

$$\epsilon_d = \frac{\lambda \epsilon_p^2}{\left(2\sqrt{2K} + 8\sqrt{\lambda}\right)^2}.$$

Proof: Theorem 2 implies that given ϵ_p it is sufficient to find an ϵ_d such that

$$4\epsilon_d + 4\sqrt{\epsilon_d}\left(\sqrt{2} + \sqrt{K/2\lambda}\right) \leq \epsilon_p. \tag{17}$$

Since the optimal value of the (P_{SVM}) criterion G_P is ≤ 1 and $G_P \geq 0$ we need only consider $\epsilon_p \leq 1$. Substituting ϵ_d into (17) and simplifying gives

$$\begin{aligned}4\epsilon_d + 4\sqrt{\epsilon_d}\left(\sqrt{2} + \sqrt{K/2\lambda}\right) &= \frac{4\lambda\epsilon_p^2}{\left(2\sqrt{2K} + 8\sqrt{\lambda}\right)^2} + \frac{\left(2\sqrt{2K} + 4\sqrt{2\lambda}\right)\epsilon_p}{2\sqrt{2K} + 8\sqrt{\lambda}} \\ &= \left(\frac{2\sqrt{\lambda}\epsilon_p}{2\sqrt{2K} + 8\sqrt{\lambda}}\right)^2 - (4 - 2\sqrt{2})\left(\frac{2\sqrt{\lambda}\epsilon_p}{2\sqrt{2K} + 8\sqrt{\lambda}}\right) + \epsilon_p \\ &= -\left(\frac{2\sqrt{\lambda}\epsilon_p}{2\sqrt{2K} + 8\sqrt{\lambda}}\right)\left(4 - 2\sqrt{2} - \frac{2\sqrt{\lambda}\epsilon_p}{2\sqrt{2K} + 8\sqrt{\lambda}}\right) + \epsilon_p \\ &\leq \epsilon_p\end{aligned}$$

where the last step follows from the fact that $\epsilon_p \leq 1$ implies $\frac{2\sqrt{\lambda}\epsilon_p}{2\sqrt{2K}+8\sqrt{\lambda}} \leq 1/4$ which in turn implies that $4 - 2\sqrt{2} - \frac{2\sqrt{\lambda}\epsilon_p}{2\sqrt{2K}+8\sqrt{\lambda}} > 0$. ■

Remark 2 *The expression for ϵ_d in Corollary 1 can often be simplified. For example if the conditions in Corollary 1 are satisfied and $\lambda \leq 1$ then*

$$\epsilon_d = \frac{\lambda\epsilon_p^2}{\left(2\sqrt{2K} + 8\right)^2}$$

is sufficient, and if $K \geq 1$ then

$$\epsilon_d = \frac{\lambda\epsilon_p^2}{121K}$$

is sufficient.

3 The L1–SVMD Learning Algorithm

In this section we describe an algorithm for the L1–SVM variant of (P_{SVM}) with $u_i = 1/n$, $i = 1,...,n$. This variant is denoted by (P_{L1-SVM}) and is used to design classifiers for the standard classification problem. We combine Simon's rate certifying algorithm for the dual [8, 24, 16] with a dual–to–primal algorithm based on Theorem 2 to form a polynomial–time learning algorithm called L1–SVMD. L1–SVMD, shown in Procedure 1, accepts a training set T, a kernel function k, a regularization parameter λ and an accuracy parameter ϵ_p, and produces an ϵ_p–minimizer $(\bar{\psi}_T, \bar{b}_T) \in \mathcal{O}_{\epsilon_p}(P_{L1-SVM})$. Note that since the dimension of $\bar{\psi}_T$ may be large (even infinite) L1–SVMD does not compute it directly; instead it returns $\bar{\alpha}_T$ which is used to implement the SVM decision function by way of $\bar{\psi}_T \cdot \Phi(x) = \frac{1}{2\lambda}\sum_{i=1}^n (\bar{\alpha}_T)_i y_i k(x_i, x)$.

The L1–SVMD algorithm has four parts; lines 3–7 determine an exact solution for the degenerate case where all data samples have the same label, lines 9–12 compute variables that define an instance of the primal and dual QP problems, lines 14–28 compute an approximate dual solution $\bar{\alpha}_T$, and line 31 computes the offset $\bar{b}_T$. The approximate dual solution $\bar{\alpha}_T$ is determined by a *rate certifying decomposition algorithm* that employs working sets W of size 2 determined by Simon's algorithm [24]. Lines 19–20 of this algorithm compute the value $\acute{m}$ which is used to determine the maximum number of iterations performed by the *main loop* in lines 22-27. The main loop updates α by solving the Wolfe Dual $(\acute{D}_{L1-SVM})$ restricted to the two components specified by the working set, updates the gradient g, uses Simon's algorithm to determine the next working set, and terminates after no more than $\lceil \acute{m} \rceil$ iterations where $\lceil \acute{m} \rceil$ is the smallest integer greater than or equal to $\acute{m}$. The condition $W^m = \emptyset$ indicates that an exact solution has been found and if this occurs for $m < \lceil \acute{m} \rceil$ then the algorithm terminates early. The value $\acute{m}$ is determined by computing the accuracy $0 < \epsilon_d \leq 1$ (on line 19) according to Remark 2 and then computing $\acute{m}$ (on line 20) to guarantee an ϵ_d–maximizer for the dual. The expression on line 20 is determined by applying Theorem 3 in Hush et al. [8] with $S = \max_i u_i = 1/n$, $L = K/2\lambda$ where $K \geq \max_i k(x_i, x_i)$, $R^* - R(\alpha^0) \leq 1$ where $R = \acute{G}_D$ is the Wolfe Dual criterion, and $\tau = 1/n$ to obtain the following corollary.

Corollary 2 *For $0 < \epsilon_d < 1$ and $K \geq \max_i k(x_i, x_i)$ the main loop of Simon's rate certifying algorithm illustrated in lines 22–27 of Procedure 1 produces an ϵ_d–maximizer of $(\acute{D}_{SVM})$ after $\lceil \acute{m} \rceil$ iterations where*

$$\acute{m} = \begin{cases} 2n \ln \frac{1}{\epsilon_d}, & \epsilon_d \geq \frac{2K}{\lambda n} \\ 2n\left(\frac{2K}{\lambda \epsilon_d n} - 1 + \max\left(0,\ \ln \frac{\lambda n}{2K}\right)\right), & \epsilon_d < \frac{2K}{\lambda n} \end{cases} .$$

Applying this corollary with $K = K_n := \max_i k(x_i, x_i)$ gives the expression for $\acute{m}$ on line 20. Once $\bar{\alpha}_T$ has been determined line 31 determines $\bar{b}_T$ by computing an exact solution to the one dimensional convex optimization problem specified in Theorem 2. To see this recall that Theorem 2 gives

$$\bar{b}_T \in \arg\min_b \sum_{i=1}^n u_i \max\left(0, 1 - y_i(\bar{\psi}_T \cdot z_i + b)\right).$$

It is easy to show that this is equivalent to line 31 by noting that

$$g_i = 1 - y_i \bar{\psi}_T \cdot z_i,\ i = 1, ..., n$$

where g_i is the i^{th} component of the gradient $g = -Q\bar{\alpha}_T + 1$.

Since the number of iterations $\lceil \acute{m} \rceil$ depends on the random variable $K_n = \max_i k(x_i, x_i)$ the run time of L1–SVMD is a random variable. To establish a deterministic run time bound for L1–SVMD we define

$$\bar{K} \; := \; \sup_{x \in X} k(x,x)$$

and restrict to kernels where $\bar{K}$ is finite. This includes the Gaussian RBF kernel $k(x,x') = e^{-\sigma^2 \|x-x'\|^2}$ for which $\bar{K} = 1$. The following is a technical lemma that is needed to establish a deterministic run time bound for L1–SVMD in terms of $\bar{K}$.

Lemma 2 *For $0 \le \acute{\epsilon} \le 1$ let $M(\acute{K}, \acute{\epsilon}) = \lceil \acute{m}(\acute{K}, \acute{\epsilon}) \rceil$ where $\acute{m}$ is the function specified in Corollary 2 with variable $(K, \epsilon_d) = (\acute{K}, \acute{\epsilon})$. If $K_1 \le K_2$ and $\epsilon_1 \ge \epsilon_2$ then $M(K_1, \epsilon_1) \le M(K_2, \epsilon_2)$.*

Proof: It is easy to verify that the function $\acute{m}$ is monotonically decreasing in $\acute{\epsilon}$ and therefore $\acute{m}(K_1, \epsilon_1) \le \acute{m}(K_1, \epsilon_2)$. It is also easy to verify that $\acute{m}$ is continuous and piecewise differentiable in $\acute{K}$. Therefore to establish that $\acute{m}$ is monotonically increasing in $\acute{K}$ it is sufficient to show that $\frac{\partial \acute{m}}{\partial \acute{K}} \ge 0$ over each of the three intervals where it is differentiable. The partial derivatives over these intervals are given by

$$\frac{\partial \acute{m}}{\partial K} = \begin{cases} 0, & K < \frac{\epsilon_d \lambda n}{2} \\ 2n \left(\frac{2}{\lambda n \epsilon_d} - \frac{1}{K} \right), & \frac{\epsilon_d \lambda n}{2} < K < \frac{\lambda n}{2} \\ \frac{4}{\lambda \epsilon_d}, & \frac{\lambda n}{2} < K \end{cases} .$$

Clearly $\frac{\partial \acute{m}}{\partial \acute{K}}$ is nonnegative over all three intervals. Thus we conclude that $\acute{m}$ is monotonically increasing in $\acute{K}$ and therefore $\acute{m}(K_1, \epsilon_2) \le \acute{m}(K_2, \epsilon_2)$. Combining the two steps gives $\acute{m}(K_1, \epsilon_1) \le \acute{m}(K_2, \epsilon_2)$ and since $\lceil \cdot \rceil$ is nondecreasing the proof is finished. ■

With the help of the above lemma we can apply Corollaries 1 and 2 with $K = \bar{K}$ to obtain a deterministic bound on the number of iterations in the main loop of Simon's decomposition algorithm. This allows us to establish the following polynomial bound on the run time of L1–SVMD.

Theorem 3 *Consider the L1–SVMD algorithm with a fixed kernel $k : X \times X \to \mathbb{R}$ where $\bar{K} = \sup_{x \in X} k(x,x)$ is finite and $\bar{K} \ge 1$. Furthermore let τ_k be an upper bound on the computation required to evaluate k. Then the L1–SVMD algorithm with inputs $T = ((x_1, y_1), \ldots, (x_n, y_n)) \in (X \times Y)^n$, $0 < \lambda \le 1$, and $0 < \epsilon_p \le 1$ produces an ϵ_p–minimizer $(\bar{\psi}_T, \bar{b}_T) \in \mathcal{O}_{\epsilon_p}(P_{L1-SVM})$ with run time*

$$\begin{matrix} O\left(\tau_k n^2 + n^2 \ln \frac{1}{\lambda \epsilon_p^2} \right), & \epsilon_p \ge 11\sqrt{2} \bar{K} \lambda^{-1} n^{-1/2} \\ O\left(\tau_k n^2 \; + \; \frac{\lambda^{-2} n}{\epsilon_p^2} \; + \; n^2 \max(0, \; \ln \lambda n) \right), & \epsilon_p < 11\sqrt{2} \bar{K} \lambda^{-1} n^{-1/2} \end{matrix} .$$

Proof: The L1–SVMD algorithm sets $\alpha^0 = 0$ (on line 15) giving an initial dual criterion value of 0 thereby guaranteeing that the final dual criterion value is nonnegative. Thus the L1–SVMD algorithm is guaranteed to produce an ϵ_p–minimizer of (P_{L1-SVM}) since it computes ϵ_d according to Remark 2, computes an ϵ_d–maximizer of the dual whose dual criterion value is nonnegative thereby satisfying the assumptions of Theorem 2, and performs the dual–to–primal map prescribed in Theorem 2. To determine the run time we start by observing that the computation in lines 3–21 is dominated by the $O(\tau_k n^2)$ computation required to produce the Q matrix. The computation of the *main loop* in lines 22–27 is $\lceil \bar{m} \rceil L$ where $\lceil \bar{m} \rceil$ is the number of iterations and L is the computation per iteration. The computation per iteration satisfies $L = O(n)$ since solving the 2–variable QP problem on line 24 takes constant time, updating the gradient on line 25 requires $O(n)$ computation (due to the sparsity of $\alpha^m - \alpha^{m-1}$), and since Simon has provided an $O(n)$ algorithm for the computation on line 26 [24]. To obtain a bound on the number of iterations let $\bar{\epsilon}_d = \frac{\lambda\epsilon_p^2}{121\bar{K}}$ be the accuracy obtained by applying Remark 2 with $K = \bar{K}$. Note that this accuracy is less than or equal to the accuracy value used to determine $\acute{m}$ in the L1–SVMD algorithm. Now apply Corollary 2 with $\epsilon_d = \bar{\epsilon}_d$ and $K = \bar{K}$ to obtain a number of iterations $\bar{m}$. Then Lemma 2 gives $\bar{m} > \acute{m}$. Substituting $\bar{\epsilon}_d$ into the expression for $\bar{m}$ from Corollary 2 and simplifying gives

$$\lceil \bar{m} \rceil L = \begin{cases} O\left(n^2 \ln \frac{1}{\lambda\epsilon_p^2}\right), & \epsilon_p \geq 11\sqrt{2}\bar{K}\lambda^{-1}n^{-1/2} \\ O\left(\frac{\lambda^{-2}n}{\epsilon_p^2} \; + \; n^2 \max\left(0, \;\ln \lambda n\right)\right), & \epsilon_p < 11\sqrt{2}\bar{K}\lambda^{-1}n^{-1/2} \end{cases} .$$

Next we prove that the computation of the main loop dominates the computation of the offset in line 31 thereby finishing the proof.

To compute the offset we must minimize the criterion $\sum_{i=1}^n u_i \max\big(0, \; y_i(g_i - b)\big)$ over b. This criterion is the sum of hinge functions with slopes $-u_i y_i$ and b–intercepts g_i. It is easy to verify that if the u_i are nonnegative, the g_i are bounded, and the y_i are not all equal then the set of optimal solutions is bounded. Furthermore, it is easy to verify that the finite set $\{g_i, i = 1, ..., n\}$ contains an optimal solution $\bar{b}_T$. The run time of the algorithm that performs a brute force computation of the criterion for every member in this set is $O(n^2)$. However this can be reduced to $O(n \log n)$ by first sorting the values g_i and then visiting them in order, using constant time operations to update the criterion value at each step. Thus, the computation required for line 31 is $O(n \log n)$ which is dominated by the computation in the main loop. ■

Remark 3 *In practice the run time of the L1–SVMD algorithm can often be improved by extending Simon's algorithm and employing a different stopping rule. For example, Hush et al. [8] introduce an extension to Simon's algorithm that possesses the same run time bounds but is shown empirically to provide substantially improved convergence rates. In addition they introduce an* `adaptive stopping rule` *that guarantees the same accuracy but is shown empirically to stop the algorithm in far fewer than $\lceil \acute{m} \rceil$ iterations.*

4 Appendix

A proof that the duality gap is zero for the finite dimensional L1–SVM primal–dual pair can be found in [5]. The following theorem extends that result to the (possibly) infinite dimensional primal–dual pair in (1)–(2).

Theorem 4 *Consider the Lagrangian*

$$L(\psi, \xi, b; \alpha, \beta) = \frac{1}{2}\|\psi\|_H^2 + u \cdot \xi + \sum_{i=1}^n \alpha_i \Big(1 - \xi_i - y_i(z_i \cdot \psi + b)\Big) - \sum_{i=1}^n \beta_i \xi_i$$

of (1) defined for $(\psi, \xi, b) \in B \; := \; H \times \mathbb{R}^n \times \mathbb{R}$ and $(\alpha, \beta) \in E_{2n} \; := \; \{\alpha, \beta : \alpha \geq 0, \beta \geq 0\}$. Then we have

$$\inf_{(\psi,\xi,b)\in B} \; \sup_{(\alpha,\beta)\in E_{2n}} L(\psi, \xi, b; \alpha, \beta) = \sup_{(\alpha,\beta)\in E_{2n}} \; \inf_{(\psi,\xi,b)\in B} L(\psi, \xi, b; \alpha, \beta).$$

Proof: Consider $(\psi_0, \xi_0, b_0) := (0, 2, 0)$ where we use the notation $2 \in \mathbb{R}^n$ for the vector $(2, 2, ..., 2, 2)$. Then since $L(\psi_0, \xi_0, b_0; \alpha, \beta) = 2 \cdot u - \alpha \cdot 1 - \beta \cdot 2$ it follows that for any value a that the set

$$\{(\alpha, \beta) \in E_{2n} : L(\psi_0, \xi_0, b_0; \alpha, \beta) \geq a\} = \{(\alpha, \beta) \in E_{2n} : \alpha \cdot 1 + \beta \cdot 2 \leq 2 \cdot u - a\}$$

is compact. Consequently, we can apply [2, Theorem 3.7] to obtain the assertion. ■

References

[1] Jose L. Balcazar, Yang Dai, and Osamu Watanabe. Provably fast training algorithms for support vector machines. In *Proceedings of the 1st International Conference on Data Mining ICDM*, pages 43–50, 2001.

[2] V. Barbu and Th. Precupanu. *Convexity and Optimization in Banach Spaces.* D. Reidel Publ., Dordrecht, 1986.

[3] Marshall Bern and David Eppstein. Optimization over zonotopes and training support vector machines. *Lecture Notes in Computer Science*, 2125:111–121, 2001.

[4] P.-H. Chen, R.-E. Fan, and C.-J. Lin. A study on SMO-type decomposition methods for support vector machines. Technical report, 2005. http://www.csie.ntu.edu.tw/∼cjlin/papers.html.

[5] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press, Canbridge ; United Kingdom, 1st edition, 2000.

[6] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II.* Springer, New York, 1996.

[7] C.-W. Hsu and C.-J. Lin. A simple decomposition algorithm for support vector machines. *Machine Learning*, 46:291–314, 2002.

[8] D. Hush, P. Kelly, C. Scovel, and I. Steinwart. QP algorithms with guaranteed accuracy and run time for support vector machines. Technical report LAUR 05-5165, Los Alamos National Laboratory, 2005. (submitted for publication).

[9] D. Hush and C. Scovel. Polynomial-time decomposition algorithms for support vector machines. *Machine Learning*, 51:51–71, 2003.

[10] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning.* MIT Press, Cambridge, MA, 1998.

[11] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, and K.R.K. Murthy. A fast iterative nearest point algorithm for support vector machine classifier design. *IEEE Transactions on Neural Networks*, 11:637–649, 2000.

[12] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, and K.R.K. Murthy. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, 13:637–649, 2001.

[13] P. Laskov. Feasible direction decomposition algorithms for training support vector machines. *Machine Learning*, 46(1–3):315–349, 2002.

[14] S.-P. Liao, H.-T. Lin, and C.-J. Lin. A note on the decomposition methods for support vector regression. *Neural Computation*, 14:1267–1281, 2002.

[15] N. List and H.U. Simon. A general convergence theorem for the desomposition method. In J. Shawe-Taylor and Y. Singer, editors, *17th Annual Conference on Learning Theory, COLT 2004, volume 3120 of Lecture Notes in Computer Science*, pages 363–377, 2004.

[16] N. List and H.U. Simon. General polynomial time decomposition algorithms. In P. Auer and R. Meir, editors, *18th Annual Conference on Learning Theory, COLT 2005*, pages 308–322, 2005.

[17] O.L. Mangasarian. *Nonlinear Programming.* SIAM Publishers, Philadelphia, PA, 1994.

[18] O.L. Mangasarian and D.R. Musicant. Successive overrelaxation for support vector machines. *IEEE Transactions on Neural Networks*, 10:1032–1037, 1999.

[19] O.L. Mangasarian and D.R. Musicant. Lagrangian support vector machines. *Journal of Machine Learning Research*, 1:161–177, 2001.

[20] E.E. Osuna, R. Freund, and F. Girosi. Support vector machines: training and applications. Technical Report AIM-1602, MIT, 1997.

[21] J.C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Scholkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 41–64. MIT Press, Cambridge, MA, 1998.

[22] R.T. Rockafellar. *Convex Analysis.* Princeton University Press, Princeton, New Jersey, 1970.

[23] C. Scovel, D. Hush, and I. Steinwart. Approximate duality. 2005. (http://ml.lanl.gov/pubs_ml.shtml, submitted for publication).

[24] H.U. Simon. On the complexity of working set selection. In *Proceedings of the 15th International Conference on Algorithmic Learning Theory*, 2004.

[25] I. Steinwart, D. Hush, and C. Scovel. A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6:211–232, 2005.

[26] I. Steinwart and C. Scovel. Fast rates for support vector machines using Gaussian kernels. Technical report, Los Alamos National Laboratory LA-UR 04-8796, 2004. http://ml.lanl.gov/pubs_ml.shtml, submitted to Annals of Statistics (2004).

[27] V. Vapnik. *Statistical Learning Theory.* John Wiley and Sons, Inc., New York, NY, 1998.

Procedure 1 The L1–SVMD Algorithm.

```
INPUTS: training set $T$, kernel $k$, regularization parameter $\lambda$, and accuracy parameter $\epsilon_p$

Compute an exact solution for degenerate case
if (all $y_i$ take the same value $\acute{y}$) then
    $\bar{\alpha}_T \leftarrow 0$, $\bar{b}_T \leftarrow \acute{y}$
    Return($\bar{\alpha}_T$, $\bar{b}_T$)
end if

Compute QP variables for the L1–SVM
$u_i = 1/n, \quad i = 1, ..., n$
$K_n \leftarrow \max_i k(x_i, x_i)$
$Q_{ij} \leftarrow y_i y_j k(x_i, x_j)/2\lambda, \quad i, j = 1, ..., n$

Compute an initial feasible point $\alpha^0$, corresponding gradient $g^0$, and initial working set $W^0$
$\alpha^0 \leftarrow 0, \quad g^0 \leftarrow 1$
$W^0 \leftarrow \{i_1, i_2\}$ where $y_{i_1} = 1, y_{i_2} = -1$

Compute an $\epsilon_d$–maximizer of the dual QP
$\epsilon_d \leftarrow \min\left(1, \frac{\lambda \epsilon_p^2}{121 K_n}\right)$
$\acute{m} \leftarrow \begin{cases} 2n \ln \frac{1}{\epsilon_d}, & \epsilon_d \geq \frac{2K_n}{\lambda n} \\ 2n\left(\frac{2K_n}{\lambda \epsilon_d n} - 1 + \max\left(0, \ \ln \frac{\lambda n}{2K_n}\right)\right), & \epsilon_d < \frac{2K_n}{\lambda n} \end{cases}$
$m \leftarrow 0$
repeat
    $m \leftarrow m + 1$
    $\alpha^m \leftarrow$ solve the 2–variable QP determined by $\alpha^{m-1}$ and $W^{m-1}$
    $g^m \leftarrow g^{m-1} - Q(\alpha^m - \alpha^{m-1})$
    $W^m \leftarrow$ use $\alpha^m$ and $g^m$ to compute next working set using Simon's algorithm
until $\big((W^m = \emptyset)$ or $(m = \lceil \acute{m} \rceil)\big)$
$\bar{\alpha}_T \leftarrow \alpha^m$

Compute the offset
$\bar{b}_T \leftarrow \arg\min_b \big(\sum_{i=1}^n u_i \max(0, g_i^m - y_i b)\big)$

Return($\bar{\alpha}_T$, $\bar{b}_T$)
```